MATLAB Simulation of Forward Planar Kinematics of a Two-Link Robot Arm Using Denavit-Hartenberg Representation

MECE 617 Activity

Ivan John A. Naparota
University of San Carlos – Talamban Campus
Department of Electrical and Electronics Engineering
ivanjohnnaparota66@gmail.com

***Abstract* — the use of kinematic equations of a robot to compute the position of the end-effecter from specified values of the joint parameters is called to be Forward Kinematics [1], a common method of solving problems of Forward Kinematics is the so called Denavit-Hartenberg Representation, it follows a four parameter representation commonly called as Denavit–Hartenberg Parameters or the "DH Parameters". DH parameters are the four parameters associated with a particular convention for attaching reference frames to the links of a spatial kinematic chain, or robot manipulator.**

This activity, as to be considered as a coursework for MECE 617, shows how to solve problems of Forward Kinematics with the use of the said method. Also, this activity exploits MATLAB to avoid manual solutions and to benefit from MATLAB's simulation prowess.

Keywords – Forward Kinematics, Denavit-Hartenberg Representation, MATLAB

I. INTRODUCTION

Denavit–Hartenberg parameters (also called DH parameters) are the four parameters associated with a particular convention for attaching reference frames to the links of a spatial kinematic chain, or robot manipulator. Jacques Denavit and Richard Hartenberg introduced this convention in 1955 in order to standardize the coordinate frames for spatial linkages [4].

The most important requirement in this convention is to be able to identify the parameters necessary for this convention, this will be defined further in the next parts of this paper. These parameters will then be used to perform the necessary operations that mainly involves multiplication of matrices for us to be able to obtain the position vector of the end effecter, which is the main concern of problems of forward kinematics.

II. EQUATIONS AND MATLAB CODES

A. *Formulas and Equations*

For a specific i, we obtain the T matrix, T = ${}^{0}A_{i}$, which specifies the position and orientation of the endpoint of the manipulator with respect to the base coordinate system. Considering T matrix to be of the form [2]

$$T = \begin{bmatrix} cos\theta & -cos\alpha sin\theta & sin\alpha sin\theta & rcos\theta \\ sin\theta & cos\alpha cos\theta & -sin\alpha cos\theta & rsin\theta \\ 0 & sin\alpha & cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

See Figure 1,

Where:

d – depth along the previous joint's z axis

θ – angle about the previous z to align its x with the new origin

r – distance along the rotated x axis

α – rotation about the new x axis to put z in its desired orientation

The homogeneous matrix ${}^{0}T_{i}$ which specifies the location of the ith coordinate frame with respect to the base coordinate system is the chain product of successive coordinate transformation matrices of ${}^{i-1}A_{i}$, and is expressed as [3]

$${}^{0}T_{i} = {}^{0}A_{1}\,{}^{1}A_{2} \ldots {}^{i-1}A_{i} \quad = \quad \begin{bmatrix} 0R_i & 0P_i \\ 0 & 1 \end{bmatrix}$$

Where:

P – Position vector of the hand

B. *MATLAB Codes*

```
clc;
clear all
```

```
close all

depth1 = 'Provide Depth for Joint1:';
depth2 = 'Provide Depth for Joint2:';
alpha1 = 'Provide Offset Angle for Joint1:';
alpha2 = 'Provide Offset Angle for Joint2:';
length1 = 'Provide Length of Link 1:';
length2 = 'Provide Length of Link 2:';
theta1 = 'Provide Angle Theta for Joint 1: ';
theta2 = 'Provide Angle Theta for Joint 2: ';
d1 = input(depth1);
d2 = input(depth2);
a1 = input(alpha1);
a2 = input(alpha2);
l1 = input(length1);
l2 = input(length2);
t1 = input(theta1);
t2 = input(theta2);

transformation1 = [cosd(t1) -(cosd(a1)*sind(t1)) sind(a1)*sind(t1) l1*cosd(t1)
sind(t1) cosd(a1)*cosd(t1) -(sind(a1)*cosd(t1)) l1*sind(t1)
0 sind(a1) cosd(a1) d1
0 0 0 1];

transformation2 = [cosd(t2) -(cosd(a2)*sind(t2)) sind(a2)*sind(t2) l2*cosd(t2)
sind(t2) cosd(a2)*cosd(t2) -(sind(a2)*cosd(t2)) l2*sind(t2)
0 sind(a2) cosd(a2) d2
0 0 0 1];

multiply = transformation1 * transformation2;
output = [multiply(1,4) multiply(2,4)]';
disp('End Effecter is Located on Point(x,y):');
disp(output);

maxaxis = l1 + l2;
line([0 (l1*cosd(t1))],[0 (l1*sind(t1))],'color','r');
line([(l1*cosd(t1)) ((l1*cosd(t1))+(l2*cosd(t1+t2)))],[(l1*sind(t1)) ((l1*sind(t1)) + (l2*sind(t1+t2)))],'color','g');
axis([-maxaxis maxaxis -maxaxis maxaxis]);
title('2-Planar Forward Kinematics via DH')
xlabel('X-Axis');
ylabel('Y-Axis')
clear all;
grid on


disp('Type [TwoLinkForwardDH] in Command Window to Try Again');
```

IV. RESULTS AND DISCUSSIONS

As shown in Figure 1, the program will ask for the DH parameters, since we are only looking at a robot arm with both links lying on the same plane, depth, and offset angle will have a value of zero.

Depth for Joint 1: 0

Depth for Joint 2: 0

Offset angle for Joint 1: 0

Offset angle for Joint 2: 0

Length of Link 1: 3 units

Length of Link 2: 2 units

Angle Theta for Joint 1: 30 degrees

Angle Theta for Joint 2: 40 degrees

Position of End-Effecter X: <u>3.2821</u>

Position of End-Effecter Y: <u>3.3794</u>

We'll get a point (x,y) which is (3.2821, 3.3794) as the position of our end effecter.

Let us check, refer to Figure 2 for the diagram,

End effecter x = 3cos(30) + 2cos(30+40) = <u>3.28211</u>

End effecter y = 3sin(30) + 2sin(30+40) = <u>3.37938</u>

So, solving through trigonometric identities, we can validate that our Denavit – Hartenberg solution is correct.

V. CONCLUSION

Denavit – Hartenberg representation is capable of solving a forward kinematics problem just like this one, the only important thing to do is to be able to identify the Denavit – Hartenberg parameters as defined above to obtain the needed transformation matrices for each joint, and the chain product of successive coordinate transformation matrices for each joints will give you the position of the end effecter.

Also, MATLAB would help a lot to perform the necessary operations without needing manual solutions.

VI. REFERENCES

1. Paul, Richard (1981). Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators. MIT Press, Cambridge, MA. ISBN 978-0-262-16082-7.

2. Fu, K. S., Gonzales, R. C., .Lee, C. S. G., ROBOTICS : Control, Sensing, Vision, and Intelligence.

3. Fu, K. S., Gonzales, R. C., .Lee, C. S. G., ROBOTICS : Control, Sensing, Vision, and Intelligence.

4. https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters

VII. FIGURES

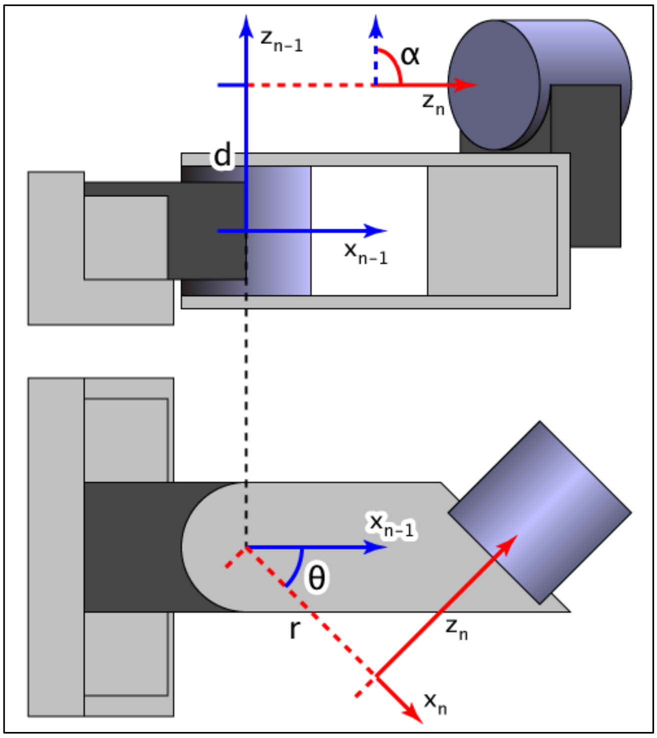

Figure 1. Denavit - Hartenberg Parameters

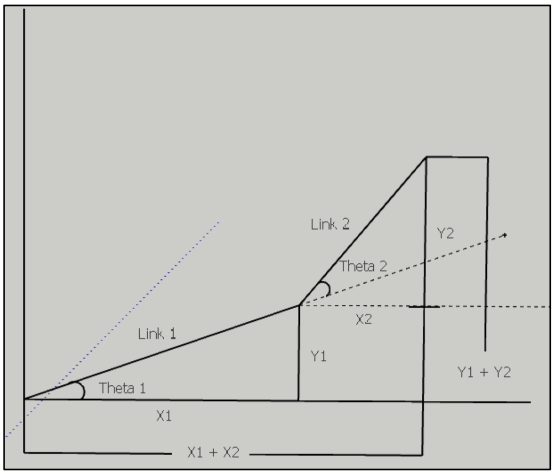

Figure 2. Two - Link Robot Arm

```
Provide Depth for Joint1:0
Provide Depth for Joint2:0
Provide Offset Angle for Joint1:0
Provide Offset Angle for Joint2:0
Provide Length of Link 1:3
Provide Length of Link 2:2
Provide Angle Theta for Joint 1: 30
Provide Angle Theta for Joint 2: 40
```

Figure 3. Inputs of the Program